C++程序设计第 4 次作业　大数据 2101 班　李嘉鹏　U202115652　　2022 年 10 月 28 日

1. 虚函数、纯虚函数可以定义为 static 成员函数吗？为什么？

不能。对于 static 成员函数，其与对象无关（参数不含 this 指针），但虚函数和纯虚函数要通过对具体对象的访问实现动态多态性，因此不能将虚函数和纯虚函数定义为 static 成员函数。

2. 构造函数、析构函数可以定义为虚函数和纯虚函数吗？为什么？

构造函数不能定义为虚函数和纯虚函数，虽然它的参数具有 this 指针，但在实际构造过程中已经确定了构造对象的类型及其内部成员的构成，不需要使用虚函数的多态特性。而析构函数可以定义为虚函数和纯虚函数，因为父类指针可以指向父类或其派生类对象，在析构时需要根据将基类析构函数定义为虚函数，通过晚期绑定确定对应的析构函数，从而避免发生内存泄漏或重复析构等问题。

3. 分析下面的程序，指出错误之处（解释错误原因），并写出 main()函数中每条正确的指令的屏幕输出结果。

```
struct A {
    int a = 0;
    int x = 1;
    void f() { cout << "A::f()"; }
    virtual void g() { cout << "A::g()"; }
    A(int x) { }
};
struct B: A {
    int x = 11;
    int y = 12;
    virtual void f() { cout << "B::f()"; }
    void g() { cout << "B::g()"; }
    void h() { cout << "B::h()"; }
    B(int x): A(x) { }
};
struct C: B {
    int x = 21;
    int y = 22;
    int z = 23;
    void f() { cout << "C::f()"; }
    void g() { cout << "C::g()"; }
    virtual void h() { cout << "C::h()"; }
    C(int x): B(x) { }
} c(1);
int main() {
    A *p = &c;
    p->f();
```

```
    p->g();
    p->h();
    cout << p->a;
    cout << p->x;
    cout << p->y;
    cout << p->z;
    /**/
    B *q = &c;
    q->f();
    q->g();
    q->h();
    cout << q->a;
    cout << q->x;
    cout << q->y;
    cout << q->z;
}
```

对于 main 函数中每条指令：

A::f()

C::g()

（p->h(); 指令有误，p 是 A 类的指针，首先在 A 类中寻找 h()，而 A 类没有定义 h()）

0

1

（cout << p->y; 指令有误，A 类中没有定义 y）

（cout << p->z; 指令有误，A 类中没有定义 z）

C::f()

C::g()

B::h()

0

11

12

（cout << q->z; 指令有误，B 类中没有定义 z）

4. 指出如下各类可访问的成员及成员的访问权限。

```
class A {
    int  a;
protected:
    int  b;
public:
    int  c;
    ~A( );
};
class B: A {
    int  a;
protected:
```

```
        A::b;
    public:
        int  c, d;
    };
    class C: protected A {
        int  a;
    protected:
        int  b, e;
    public:
        int  g;
        A::c;
    };
    struct D: B, C {
        int a;
    protected:
        int b, f;
    public:
        int e, g;
};
```

A 类：

private：a

protected：b

public：c，~A()

B 类：

private：A::c，A::~A()，a

protected：A::b

public：c，d

C 类：

private：a

protected：A::b，A::~A()，b，e

public：g，A::c

D 类：

private：a

protected：A::b，A::~A()，C::b，C::e，b，f

public：B::c，B::d，C::g，A::c，e，g

5. 指出如下程序的错误之处及其原因：

```
class A {
        int x;
        virtual int f( ) { return 0; }
        virtual int g( ) = 0;
protected:
        int  y;
```

```
public:
    virtual A( ) { }
} a;
struct B: A {
      A::x;
      using A::y;
      long f( )  { return 1L; };
      int g(int) { return 1; }
} b;
A *p = new A;
B *q = new B;
int f(A, B);
A g(B &);
int h(B *);
```

①A 类的构造函数不能定义为 virtual；

②A 类中含有纯虚函数，为抽象类，不能产生对象；

③B 继承 A 时访问不到 A::x（在 A 中权限为 private），无法改变 A::x 在 B 中的访问权限；

④B 类中 long f()与 A 类中 int f()返回值类型不同，不满足虚函数定义的条件；

⑤B 类继承了 A 类的纯虚函数 A::g()，为抽象类，同样不能产生对象；

⑥new A、new B 同理错误，A 类和 B 类均不可产生对象。

6. 指出如下程序中main()中每行语句的输出结果。

```
struct A { A( ) { cout << 'A'; } };
struct B { B( ) { cout << 'B'; } };
struct C: A { C( ) { cout << 'C'; } };
struct D: A, virtual B { D( ) { cout << 'D'; } };
struct E: A, virtual B, virtual C {
      D d;
      E( ) { cout << 'E'; }
};
struct F: A, virtual B, virtual C, D, E {
      C  c;
      E  e;
      F( ) { cout << 'F'; }
};
void main(void)
{
    A  a;
    B  b;
    C  c;
    D  d;
    E  e;
```

```
    F  f;
}
```

main()中每行语句的输出结果是：

A

B

AC

BAD

BACABADE

BACAADABADEACBACABADEF